Mapping the magnetic field using a magnetometer array with noisy input Gaussian process regression

Edridge, Thomas; Kok, Manon

DOI
10.23919/FUSION52260.2023.10224116

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 26th International Conference on Information Fusion, FUSION 2023

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Mapping the magnetic field using a magnetometer array with noisy input Gaussian process regression

Thomas Edridge and Manon Kok

Delft Center for Systems and Control,

Delft University of Technology, Delft, the Netherlands

Email: {t.i.edridge, m.kok-1}@tudelft.nl

Abstract—Ferromagnetic materials in indoor environments give rise to disturbances in the ambient magnetic field. Maps of these magnetic disturbances can be used for indoor localisation. A Gaussian process can be used to learn the spatially varying magnitude of the magnetic field using magnetometer measurements and information about the position of the magnetometer. The position of the magnetometer, however, is frequently only approximately known. This negatively affects the quality of the magnetic field map. In this paper, we investigate how an array of magnetometers can be used to improve the quality of the magnetic field map. The position of the array is approximately known, but the relative locations of the magnetometers on the array are known. We include this information in a novel method to make a map of the ambient magnetic field. We study the properties of our method in simulation and show that our method improves the map quality. We also demonstrate the efficacy of our method with experimental data for the mapping of the magnetic field using an array of 30 magnetometers.

Index Terms—Gaussian process, magnetic field, mapping, noisy inputs, sensor array.

I. INTRODUCTION

Ferromagnetic materials are prevalent in indoor environments and give rise to disturbances in the ambient magnetic field [1]–[3]. In recent years, the disturbances in magnetic field maps are used to improve localisation in indoor environments [4]–[9]. We make use of a Gaussian process (GP) [10] to learn magnetic field maps using magnetometers, which measure the ambient magnetic field. We consider the magnetometer locations as the inputs of the GP and the magnitude of the magnetometer measurements as the outputs of the GP. However, the true position of the magnetometer is frequently only approximately known, which reduces the quality of the map. In this paper, we address this shortcoming by using an array of magnetometers. The position of the array is approximately known, but the relative locations of the magnetometers on the array are assumed to be known. The information that magnetometers are placed on an array is extra information that can improve the quality of the magnetic field map [11].

Typically, GP approaches assume deterministic, known inputs. In the top of Fig. 1 we see a GP in green which assumes deterministic inputs but uses noisy inputs. This prediction is wrongfully overconfident. We are interested in the case when input locations are only approximately known. Exact inference with noisy inputs is generally intractable [12]–[15]. One solution uses an approximate analytical approach where the input noise is treated as a function of the input location
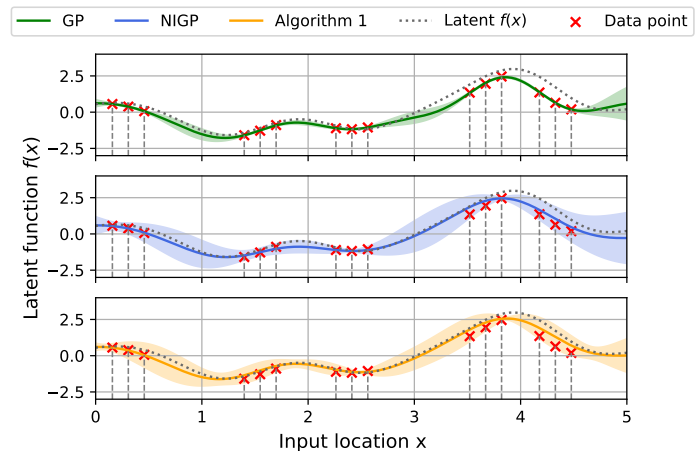

Fig. 1. 1D example of a synthetic data set where the data is collected using a noisy input array with 3 sensors. The vertical dashed grey lines show the separation of the 3 sensors on the array. We show the regression of a GP (green), NIGP (blue) and our method (orange), all with mean and a confidence interval of 95%. Data are in red and the latent function $f(x)$ is in grey. The input noise is sampled from a covariance of $\Sigma_x = 0.05$.

[16]. However, this method does not take advantage of the shape of the posterior function [17]. Knowledge of the shape of the posterior function was included in [18] for noisy prediction locations, but not for noisy training data. In noisy input GP regression (NIGP) [19] knowledge of the shape of the posterior function for both training data and prediction locations is included. To be more precise, NIGP maps the input uncertainty to the output via the gradient of the posterior mean. This is illustrated in blue in Fig. 1. When the posterior mean function is flat, no input uncertainty gets propagated. When the posterior mean function is steep, the input uncertainty has a large effect. This can be seen in the posterior variance change from GP (green) to NIGP (blue) when $x = 3$.

We show how we can build magnetic field maps using an array of magnetometers where the array location is only approximately known, but the relative locations of the magnetometers on the array are assumed to be known. To this end, we extend on the work from [19] and inform our model of the correlation between input locations arising from the fact that the magnetometers are placed on an array. Our method explicitly takes into account that the three sensors in Fig. 1 are placed on an array. This decreases the posterior variance of

our method compared to NIGP, which can be seen in the range from $x = 3$ to $x = 5$. Additionally, the posterior mean of our method more closely resembles the latent function, which can be seen around when $x = 2$ and $x = 4.5$. We show under which conditions the posterior mean function improves and the posterior variance reduces.

II. PROBLEM FORMULATION

We use an array of magnetometers to learn a map of the ambient magnetic field. The GP takes the magnetometer locations $\bar{\mathbf{x}}_{i,k}^{\mathrm{n}}$ as inputs in the navigation frame n. The array moves about in the navigation frame and this frame is fixed with the rotation of the earth. The magnitude of the magnetometer measurements $y_{i,k}$ are taken as outputs. We only have access to the noisy magnetometer locations as inputs

$$\mathbf{x}_{i,k}^{\mathrm{n}} = \bar{\mathbf{x}}_{i,k}^{\mathrm{n}} + \mathbf{e}_{\mathrm{x}}^{\mathrm{n}}, \quad \mathbf{e}_{\mathrm{x}}^{\mathrm{n}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathrm{x}}), \tag{1}$$

where we have $i = 1, \ldots, m$ magnetometers, $k = 1, \ldots, N$ time steps and $\mathbf{x}_{i,k} \in \mathbb{R}^3$, $\mathbf{e}_{i,k} \in \mathbb{R}^3$. We consider a zero mean GP and kernel function $\kappa(\cdot, \cdot) : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$

$$\begin{aligned} y_{i,k} &= f(\bar{\mathbf{x}}_{i,k}^{\mathrm{n}}) + e_{\mathrm{y}}, \quad e_{\mathrm{y}} \sim \mathcal{N}(0, \sigma_{\mathrm{y}}^2), \\ f(\cdot) &\sim \mathcal{GP}(\mathbf{0}, \kappa(\cdot, \cdot)), \end{aligned} \tag{2}$$

where $f(\cdot)$ is the latent function and e_{y} is signal noise. We construct the covariance matrix with the widely used squared exponential (SE) kernel:

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_{\mathrm{f}}^2 \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2l^2}\right). \tag{3}$$

The hyper-parameters of this kernel are the length-scale l and the signal variance σ_{f}^2. To interpolate or extrapolate a magnetic field map, we can use the GP to make predictions at previously unseen locations $\mathbf{x}_\star$, where we assume $\mathbf{x}_\star$ is deterministic. To this end, we construct kernel matrices $K(X, X) \in \mathbb{R}^{mN \times mN}$, $K(X, X_\star)$ and $K(X_\star, X_\star)$ by evaluating (3) for all pairs of data and prediction locations. The predictive posterior is normally distributed with predictive mean $\boldsymbol{\mu}_\star$ and covariance $\Sigma_{\star\star}$ as

$$\begin{aligned} \mathbf{f}_\star &\sim \mathcal{N}(\boldsymbol{\mu}_\star, \Sigma_{\star\star}), \\ \boldsymbol{\mu}_\star &= K_\star^T (K + \sigma_{\mathrm{y}}^2 \mathcal{I}_{mN})^{-1} \mathbf{y}, \\ \Sigma_{\star\star} &= K_{\star\star} - K_\star^T (K + \sigma_{\mathrm{y}}^2 \mathcal{I}_{mN})^{-1} K_\star. \end{aligned} \tag{4}$$

Where we make use of the shorthand notation $f(\mathbf{x})$ as f and $K(X, X), K(X, X_\star)$ and $K(X_\star, X_\star)$ as $K, K_\star$ and $K_{\star\star}$ respectively. As mentioned in (1), we only have access to the noisy inputs, so we substitute the true inputs with the noisy version $\bar{\mathbf{x}}_{i,k}^{\mathrm{n}} = \mathbf{x}_{i,k}^{\mathrm{n}} - \mathbf{e}_{\mathrm{x},k}^{\mathrm{n}}$. As we can see in (4), the posterior depends non-linearly on true input locations. We cannot solve this in closed form and we will make use of NIGP [19] for an approximate-analytical solution.

III. MAGNETOMETER ARRAY GP REGRESSION

A. Magnetometer array modelling assumptions

The locations of magnetometers $i = 1, \ldots, m$ relative to the centre of the array are assumed to be deterministic and are constant in the body frame b. The body frame is fixed to the centre of the array. The magnetometer locations in the navigation frame can be found at every time step k as

$$\mathbf{x}_{i,k}^{\mathrm{n}} = \bar{\mathbf{x}}_{c,k}^{\mathrm{n}} + R_k^{\mathrm{nb}} \boldsymbol{s}_i^{\mathrm{b}} + \mathbf{e}_{\mathrm{x}}^{\mathrm{n}}, \tag{5}$$

where $\mathbf{e}_{\mathrm{x}}^{\mathrm{n}}$ is defined in (1), $\bar{\mathbf{x}}_{c,k}^{\mathrm{n}} \in \mathbb{R}^3$ is the true location of the array's centre, $\boldsymbol{s}_i^{\mathrm{b}} \in \mathbb{R}^3$ is the translation from the centre of the array to magnetometer i and R^{nb} denotes a rotation from the body frame to the navigation frame. We define the matrix $H \in \mathbb{R}^{3m \times 3}$ as

$$H = \mathbf{1}_m \otimes \mathcal{I}_3, \tag{6}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{1}_m$ is a vector of ones of size m and H contains the information on how the input locations are correlated. We take all input locations on the array into the vector for a single time step $\bar{X}_k^{\mathrm{n}} = \left[(\bar{\mathbf{x}}_{1,k}^{\mathrm{n}})^{\mathsf{T}} \quad \ldots \quad (\bar{\mathbf{x}}_{m,k}^{\mathrm{n}})^{\mathsf{T}}\right]^{\mathsf{T}} \in \mathbb{R}^{3m}$. We also define the vector $S_k^{\mathrm{n}} \in \mathbb{R}^{3m}$, which contains all the relative magnetometer locations from the array's origin in the navigation frame and is defined as $S_k^{\mathrm{n}} = \left[(R_k^{\mathrm{nb}} \boldsymbol{s}_1^{\mathrm{b}})^{\mathsf{T}} \ldots (R_k^{\mathrm{nb}} \boldsymbol{s}_m^{\mathrm{b}})^{\mathsf{T}}\right]^{\mathsf{T}}$. We now have a formulation for the magnetometer locations in the normal frame in terms of the noisy array's centre location:

$$\begin{aligned} X_k^{\mathrm{n}} &= \mathbf{1}_N \otimes \bar{\mathbf{x}}_{c,k}^{\mathrm{n}} + S_k^{\mathrm{n}} + \mathbf{1}_N \otimes \mathbf{e}_{\mathrm{x}}^{\mathrm{n}}, \\ &= H\bar{\mathbf{x}}_{c,k}^{\mathrm{n}} + S_k^{\mathrm{n}} + H\mathbf{e}_{\mathrm{x}}^{\mathrm{n}}, \\ &= \bar{X}_k^{\mathrm{n}} + H\mathbf{e}_{\mathrm{x}}^{\mathrm{n}} \end{aligned} \tag{7}$$

where X_k^{n} and $\bar{X}_k^{\mathrm{n}}$ are vectors with all the noisy and true magnetometer locations respectively. The previous equation shows that the magnetometer locations for a single time step are correlated as $X_k^{\mathrm{n}} \sim \mathcal{N}(\cdot, H\Sigma_{\mathrm{x}}H^{\mathsf{T}})$.

B. Constructing GP maps using a magnetometer array

We return to (1) and replace the true input locations with the correlated input locations from (7) and our magnetometer measurements to obtain the GP model

$$\mathbf{y}_k = f(X_k^{\mathrm{n}} - H\mathbf{e}_{\mathrm{x}}^{\mathrm{n}}) + \mathbf{e}_{\mathrm{y}}, \quad \mathbf{e}_{\mathrm{y}} \sim \mathcal{N}(0, \sigma_{\mathrm{y}}^2 \mathcal{I}_m). \tag{8}$$

where we have taken all the measurements from the array into the vector $\mathbf{y}_k = \begin{bmatrix} y_{1,k} & \ldots & y_{m,k} \end{bmatrix}^{\mathsf{T}} \in \mathbb{R}^m$ from which we subtract the mean, as we use a GP with a zero-mean function. This results in the output of our GP being the disturbances in the ambient magnetic field. A closed form solution to the posterior is intractable in (4) as the posterior depends non-linearly on the input locations. We find an approximate solution by taking the first-order Taylor expansion around the input locations similar to [19]. First, we introduce $\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \ldots & \frac{\partial f(\mathbf{x})}{\partial x_d} \end{bmatrix}^{\mathsf{T}}$ and $\Delta f(X) = \mathrm{blockdiag}(\nabla f(\mathbf{x}_1), \ldots, \nabla f(\mathbf{x}_m))$ for m input locations. The approximate measurement equation is then found as

$$\mathbf{y}_k \approx f(X_k^{\mathrm{n}}) - \Delta f(X_k^{\mathrm{n}})^{\mathsf{T}} H\mathbf{e}_{\mathrm{x}}^n + \mathbf{e}_{\mathrm{y}}. \tag{9}$$

The first-order Taylor approximation allows us to find an expression for the posterior in closed form. By taking the first-order Taylor expansion, we treat the input noise as heteroskedastic output noise by propagating it to the output via the posterior gradient. This also introduces a new variable, the gradient of the posterior mean function which we need to find. We may find the posterior mean gradient for all time steps as

$$\nabla f(X_{1:N}^{\mathrm{n}}) = \nabla K^{\mathsf{T}}(K + \sigma_y^2 \mathcal{I}_n + \Sigma(X_{1:N}^{\mathrm{n}}))^{-1} \mathbf{y}_{1:N}. \quad (10)$$

Where we make use of $X_{1:N}^{\mathrm{n}} = \left[(X_1^{\mathrm{n}})^{\mathsf{T}} \quad \dots \quad (X_N^{\mathrm{n}})^{\mathsf{T}}\right]^{\mathsf{T}}$, $\mathbf{y}_{1:N} = \left[\mathbf{y}_1^{\mathsf{T}} \quad \dots \quad \mathbf{y}_N^{\mathsf{T}}\right]^{\mathsf{T}}$ and the derivative ∇K of the SE kernel can be found in [12]. The posterior gradient can then be used to find the term $\Sigma(X_{1:N}^{\mathrm{n}})$, which is an additional covariance term introduced by the input noise as

$$\Sigma(X_{1:N}^{\mathrm{n}}) = \Delta f(X_{1:N}^{\mathrm{n}})^{\mathsf{T}} \left(\mathcal{I}_N \otimes \left(H \Sigma_{\mathrm{x}} H^{\mathsf{T}}\right)\right) \Delta f(X_{1:N}^{\mathrm{n}}). \quad (11)$$

Estimating the posterior mean derivative is a chicken-and-egg problem. The posterior mean derivative (10) depends on the additional input uncertainty (11) which depends on the posterior mean derivative (10). We can find a solution by setting the posterior mean derivative to zero, which is effectively the same as doing regular GP regression. We then iterate between finding (10) and (11) until a stopping criterion is met. We use the criterion

$$\epsilon_j = ||\nabla f(X_{1:N}^{\mathrm{n}})_j - \nabla f(X_{1:N}^{\mathrm{n}})_{j-1}|| < \delta_\epsilon, \quad (12)$$

where j denotes the iteration number. We then have the predictive posterior mean and covariance as

$$\bar{\mathbf{f}}_\star \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_\star, \bar{\Sigma}_{\star\star}),$$
$$\bar{\boldsymbol{\mu}}_\star = K_\star^T \left(K + \sigma_y^2 \mathcal{I}_{mN} + \Sigma_x(X_{1:N}^{\mathrm{n}})\right)^{-1} \mathbf{y}_{1:N}, \quad (13)$$
$$\bar{\Sigma}_{\star\star} = K_{\star\star} - K_\star^T \left(K + \sigma_y^2 \mathcal{I}_{mN} + \Sigma_x(X_{1:N}^{\mathrm{n}})\right)^{-1} K_\star.$$

We summarize our method in Algorithm 1.

Algorithm 1 Posterior mean and posterior covariance estimates using a GP with correlated noisy inputs.

Inputs: $X_{1:N}^{\mathrm{n}}, \mathbf{y}_{1:N}$
Outputs: $\bar{\boldsymbol{\mu}}_*, \bar{\Sigma}_{**}$
Set $\nabla f(X_{1:N}^{\mathrm{n}}) \leftarrow 0$
while Not converged according to (12) **do**
    Evaluate $\Sigma(X_{1:N}^{\mathrm{n}})$ from (11).
    Evaluate $\nabla f(X_{1:N}^{\mathrm{n}})$ from (10).
end while

IV. ANALYSIS

We study the properties of our method compared against NIGP and the effects on the quality of the magnetic field map. First, we study the effects of the input uncertainty w.r.t. the length-scale in Section IV-A. Subsequently, in Section IV-B we study the effect of the length of the array w.r.t. the length-scale. Finally, we study the case when all measurements are from a single array in Section IV-C.

The effect of the additional input uncertainty term depends on the spatial variation in the data. In the case where there is no spatial variation in the data, the latent function is flat and the derivative is zero. In this case, the additional input uncertainty term has no effect, and Algorithm 1 and NIGP are identical to a regular GP from (4). Another case is when all magnetometers are on a flat part of the function except for one magnetometer. In this case, Algorithm 1 is identical to NIGP.

A. Effect of the input uncertainty on the quality of the map

We run 100 Monte Carlo (MC) simulations where we sample the data from the SE kernel (3) with hyper-parameters $l = 0.5, \sigma_f^2 = 1, \sigma_y^2 = 0.001$. We set the input space to $X \in [0, 5]$ and we place 60 array centre locations on a grid over the entire input domain. We use an array with 5 sensors equidistantly placed to get a total of 300 input locations. We sample noise according to (7), using a variable input covariance ranging from $\Sigma_{\mathrm{x}} = 0$ to $\Sigma_{\mathrm{x}} = 0.25$ and add this to the true input locations after which measurement noise is added according to (8).
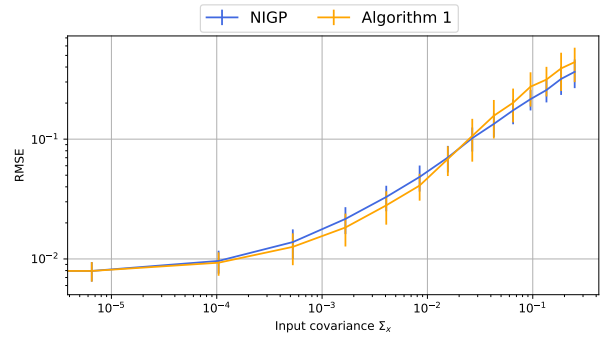

Fig. 2. The average and standard deviation of the RMSE over 100 different data sets of Algorithm 1 (orange) and NIGP (blue) as a function of the magnitude of the input uncertainty Σ_{x}.

We compare Algorithm 1 against NIGP and we plot the mean and standard deviation of the RMSE of the 100 different data sets in Fig. 2. We show how the average RMSE grows with the input covariance Σ_{x}. As the input covariance increases, from 0 to 0.02, Algorithm 1 has a smaller RMSE than NIGP. We can see that Algorithm 1 provides a better estimate of the posterior mean function compared to NIGP. This makes sense as NIGP omits the correlations between input locations. However, when the input uncertainty grows too large, Algorithm 1 performs worse than NIGP. This can be seen as the input covariance grows larger than 0.02, where the RMSE of Algorithm 1 is compared to NIGP. The reason for this is that when the input error grows too large, we cannot expect the first-order Taylor approximation to be accurate. By introducing correlations between the input locations, Algorithm 1 has fewer degrees of freedom compared to NIGP and thus NIGP is more flexible in modelling the posterior mean resulting in a lower RMSE.

Both Algorithm 1 and NIGP iterate to find the additive input covariance $\Sigma(X_{1:N}^{\mathrm{n}})$ as described in Section III-B. To illustrate
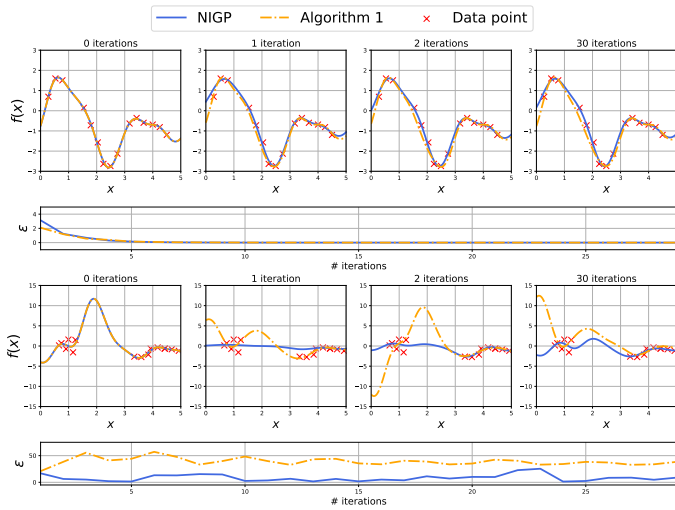
Fig. 3. Synthetic data showing 0, 1, 2 and 30 iterations of NIGP and Algorithm 1 along with the error ϵ from (12). The top row has a small covariance $\Sigma_x = 0.01$ w.r.t. $l = 0.5$ and the bottom row has a large input covariance $\Sigma_x = 0.5$.

the effect of the input error on convergence, we compare Algorithm 1 to NIGP both with a small input covariance of $\Sigma_x = 0.01$ and a large input covariance of $\Sigma_x = 0.5$ and a length-scale of $l = 0.5$ for 0, 1, 2 and 30 iterations in Fig. 3. It can be seen that for $\Sigma_x = 0.01$, both methods converge within 5 iterations. For $\Sigma_x = 0.5$, both Algorithm 1 and NIGP do not converge in the 30 iterations. The reason for this is that the input uncertainty is too large w.r.t. length-scale. Data points from different locations start to overlap. This can be seen in the bottom half of the figure around $f(x) = 0, x = 1$. This overlap in data does not adhere to the GP smoothness assumptions, causing Algorithm 1 and NIGP to not converge. NIGP is less affected by the overlap in data points as it assumes no correlations between inputs. It can be observed that NIGP remains relatively flat compared to Algorithm 1, which results in a lower error as data is sampled from a GP with zero-mean assumption.

B. Effect of the array length on the map quality

We also study the effect of the length of the array on the quality of the map. When we increase the length of the array, more spatial variation of the latent function is captured by the array. We run the simulation with the same settings as in Section IV-A. We set the input covariance to $\Sigma_x = 0.05$ and vary the length of the array from 0 to 2. We plot the RMSE and the posterior predictive variance in Figs. 4 and 5, respectively. In Fig. 4 we observe that if the array is of length 0, i.e. the sensors of the array are placed in the same location, Algorithm 1 has a smaller RMSE compared to NIGP. The reason for this is that NIGP omits the correlation terms between input locations, treating each measurement as separate, leading NIGP to be overconfident. For an array length of 0.5 (the size of the length-scale), both Algorithm 1 and NIGP have an RMSE of 0.064 and 0.048 respectively.

Algorithm 1 improves more compared to NIGP. The reason for this is that when the length of the array is bigger, Algorithm 1 is able to capture more spatial information. If we increase the array length to 2, the RMSE of Algorithm 1 and NIGP decrease to 0.030 to 0.044 respectively. The decrease in RMSE corresponds to improving the magnetic field map quality. The bigger the size of the array, the better the map quality.
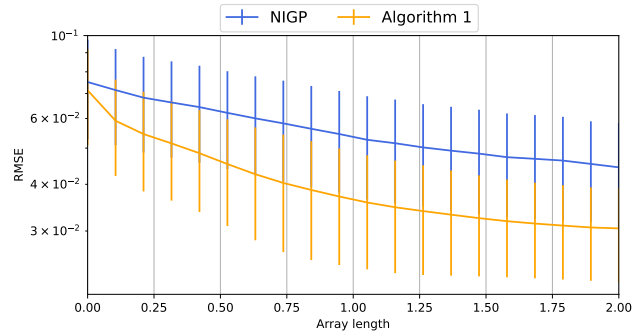

Fig. 4. Average RMSE for a simulation of 100 synthetic data sets where we vary the length of an array of 5 sensors w.r.t the length-scale l. The 5 sensors are placed equidistantly along the length of the array. We plot the average predictive posterior RMSE and its standard deviation of NIGP (blue) and Algorithm 1 (orange).
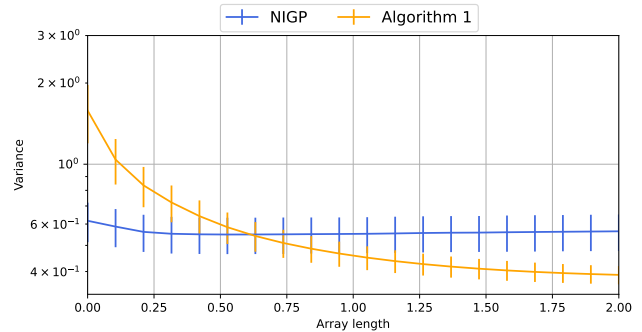

Fig. 5. Average posterior variance of a simulation of 100 synthetic data sets where we vary the length of an array of 5 sensors w.r.t the length-scale l. We plot the average and standard deviation of the predictive posterior variance of NIGP (blue) and Algorithm 1 (orange).

It can be seen in Fig. 5 that when the array length is zero, Algorithm 1 has a larger variance than NIGP. This suggests that NIGP overfits as the variance is smaller but the RMSE is bigger compared to Algorithm 1. The variance of Algorithm 1 and NIGP intersect around an array size of 0.63, just bigger than the length-scale $l = 0.5$. When the array length is 2, the variance of Algorithm 1 decreases to 0.38. If the array size is larger than the length-scale, NIGP starts to underfit as the RMSE is larger and the predictive posterior variance is also larger compared to Algorithm 1. Overall, we see that when the array length increases, the posterior variance deceases. This in line with the results shown in [20], where they derive the Cramér-Rao lower bound for estimating the position. They show that the positional error decreases as the size of the magnetometer array increase.

C. Effects when all measurements are from a single array

To illustrate the extreme case, we create a data set where all the measurements are from a single array. We omit measurement noise and add an input error of $\mathbf{e}_x = 0.5$.
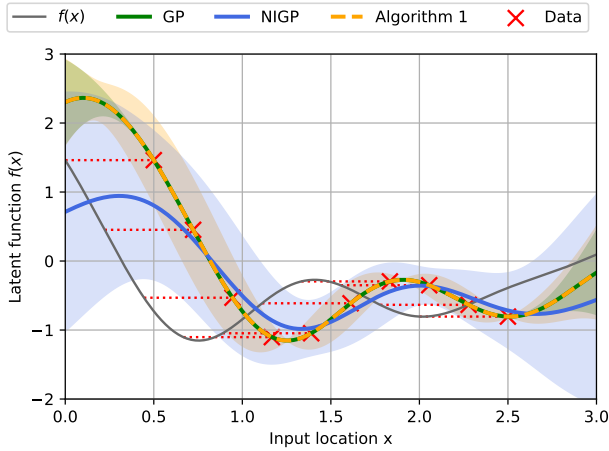

Fig. 6. Synthetic data where 10 measurements are taken with a single array. We plot the posterior mean of a regular GP (green), NIGP (blue) and Algorithm 1 (orange) each with 95% confidence interval.

In Fig. 6 we have 10 data points all captured with a single array. We can see that both Algorithm 1 and a regular GP (which assumes deterministic inputs but uses noisy inputs) are identical in their posterior mean function and they are able to capture the spatial variation of the latent function. In fact, over the span of the array, from $x = 0.5$, to $x = 2.5$, the posterior mean functions are identical to the latent function shifted in the input direction with the input error e_x.

V. EXPERIMENTS

Experimental data is collected using an array of size $0.32\text{m} \times 0.22\text{m}$ with 30 magnetometers placed on a grid as shown in Fig. 7. The ground truth position of the array and its orientation over time are measured using an optical motion capture system (OptiTrack).
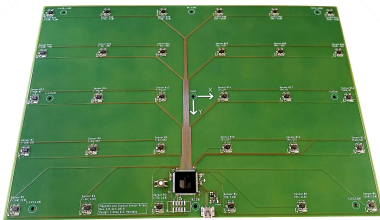

Fig. 7. Magnetometer array of dimensions $0.32\text{m} \times 0.22\text{m}$ with 30 magnetometers used in the collection of the two data sets. The magnetometers are RM3100 from PNI running at 500Hz.

Two data sets are collected using the array. In the first data set the array moves in a small square (of approximately size $3\text{m} \times 3\text{m}$). In the second data set in a big square (of approximately size $4\text{m} \times 4\text{m}$). Because the computational

complexity of GP regression scales cubically with the number of data points, we downsample both data sets to 50 smaller subsets each. Each subset has 1200 data points. In Fig. 8 we show an example of one subset for data set 1 and 2.
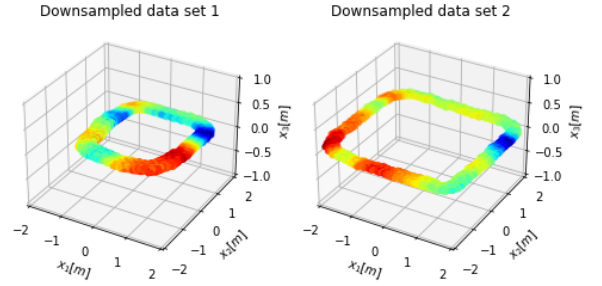

Fig. 8. Subsets of data set 1 and 2 showing 1200 measurements of a magnetometer array, where a magnetometer array moves close to the ground in a small square (left) and a big square (right). The colour indicates the value of the measurement.

A. Magnetometer calibration

To find the outputs of our GP model, we first calibrate each magnetometer for their bias (offset) $\boldsymbol{b}_i$ and distortion matrix D_i, where $i = 1, \ldots, m$ according to the initial parameter estimate of [21]. This ensures that all magnetometers would provide the same measurements of the ambient magnetic field if they would be placed at exactly the same location. We then consider the magnitude of the calibrated magnetometer measurements $y_{i,k}$ as the output of our GP model

$$y_{i,k} = || D_i^{-1} \left(\hat{\mathbf{y}}_{i,k} - \mathbf{b}_i \right) ||_2^2, \tag{14}$$

where $\hat{\mathbf{y}}_{i,k}$ are the magnetometer measurements. We consider $y_{i,k}$ as the noisy measurements of the underlying magnetic field. We then subtract the mean from all the measurements.

B. Results

We find the hyper-parameters of the GP for the two data sets by minimising the negative log marginal likelihood according to [10] using the ground truth position and the measurements of a single magnetometer. The hyper-parameters are found to be $l = 5.59 \times 10^{-1}, \sigma_f^2 = 4.90 \times 10^{-3}, \sigma_y^2 = 5.31 \times 10^{-4}$ and $l = 5.51 \times 10^{-1}, \sigma_f^2 = 4.93 \times 10^{-3}, \sigma_y^2 = 3.89 \times 10^{-4}$ for data sets 1 and 2, respectively.

For each of the 50 subsets of 1200 data points, we individually sample input errors for each array location according to (7). We sample input errors with a covariance of $\Sigma_x = \mathcal{N}(\mathbf{0}, \text{diag}(\beta, \beta, 0))$, where β is a scaling factor that ranges from 0.01 to 0.2. We don't add input noise to the vertical axis as we, primarily, move along the two horizontal axes as shown in Fig. 8.

We compare Algorithm 1 and NIGP to a reference GP that uses the true input locations, which acts as a baseline. On the left side in Fig. 9 we show the average RMSE of the 50 subset for data set 1 and 2. We can see that when $\beta = 0.01$, the average RMSE of Algorithm 1 is the lowest for both data
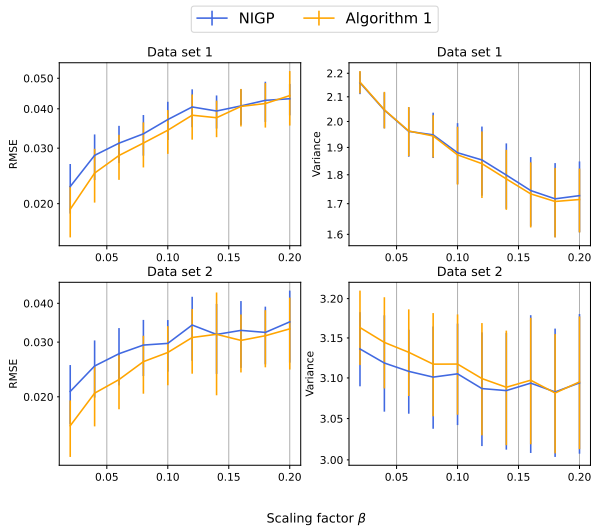
Fig. 9. Average RMSE (left) and trace of the posterior variance (right) of a simulation of the the 50 subsets for both data set 1 (top) and data set 2 (bottom). The RMSE is the error from a GP using the true input locations. We compare NIGP (blue) and Algorithm 1 (orange) all of which take noisy inputs. Input noise is sampled for each MC simulation from $\Sigma_x = \mathcal{N}(\mathbf{0}, \mathrm{diag}(\beta, \beta, 0))$, where β is a variable scaling factor.

sets. As β grows to 0.2, the RMSE of Algorithm 1 gets closer to that of NIGP. For data set 1 the RMSE of Algorithm 1 and NIGP of equal magnitude at $\beta = 0.16$ and for data set 2 they become similar from $\beta = 0.14$. This is in line with our results from Fig. 2 where we show that Algorithm 1 performs better than NIGP if the error is small enough w.r.t. to the length-scale.

On the right side in Fig. 9 we show the average trace of the posterior predictive variance of the two data sets. Interestingly, we can see that the variance decreases as we increase β. As shown in Fig. 8, we have collected the data along a fixed path. By increasing the input noise, the noisy input locations become more scattered across our prediction domain (i.e. away from the fixed path). This results in the GPs wrongfully lowering the predictive variance in places where in reality no measurement was taken. We also see that the variance of Algorithm 1 has approximately the same magnitude as NIGP. We saw a similar effect in Fig. 5. The array is of size $0.32\mathrm{m} \times 0.22\mathrm{m}$ where the length-scale of size $l \approx 0.5$ for both data sets. To decrease the variance compared to NIGP, we need an array of size larger than the length-scale.

In Fig. 10 we illustrate the difference in map quality between Algorithm 1 (right) and NIGP (middle) compared to a GP using the true input locations (left). Around locations $\mathbf{x} = \begin{bmatrix} 0 & -2 \end{bmatrix}^{\mathsf{T}}$ and $\mathbf{x} = \begin{bmatrix} -2 & 1 \end{bmatrix}^{\mathsf{T}}$ Algorithm 1 looks more similar to the reference map compared to NIGP. This shows that a lower RMSE as shown in Fig. 9 results in a higher map quality.

VI. Conclusion and future work

In this paper, we show that by using an array of magnetometers, we can increase the quality of the magnetic field map.
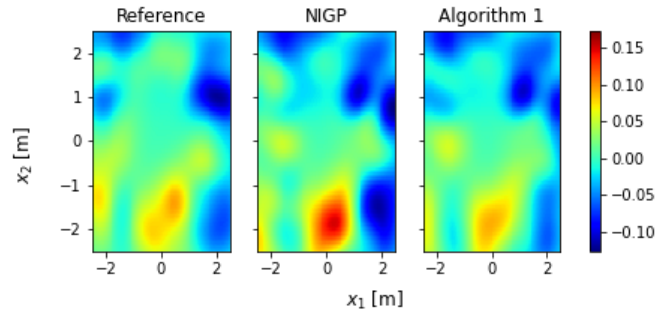

Fig. 10. Visual example of the map quality of one of the downsampled data sets from data set 2. The absolute value of the disturbances in the local magnetic field is plotted over an area of $5 \times 5\mathrm{m}^2$. NIGP (middle) and Algorithm 1 (right) are compared to a GP using the true input locations (left) as a reference map. The input error is sampled from $\Sigma_x = \mathcal{N}(\mathbf{0}, \mathrm{diag}(0.05, 0.05, 0))$.

Extending on NIGP [19], our method uses the information that the magnetometers are placed on an array. We show that compared to NIGP, our method decreases the RMSE (in comparison to the true latent function) of the predictive posterior mean if the input error is small enough w.r.t. the length-scale. We also show that if the array size is small w.r.t. the length-scale, our method compensates by increasing the posterior variance compared to NIGP. If the array size is big w.r.t. the length-scale, our method reduces the variance compared to NIGP. Finally, we show that our method improves the map quality using experimental data set with an array of 30 magnetometers. An interesting direction of future work is how our method can become more scalable whilst preserving the correlated covariance structure.

References

[1] E. Le Grand and S. Thrun. 3-axis magnetic field mapping and fusion for indoor localization. In *Proceedings of the 2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 358–364. IEEE, 2012.

[2] M. Kok and A. Solin. Scalable magnetic field SLAM in 3D using Gaussian process maps. In *Proceedings of the 21st international conference on information fusion (FUSION)*, pages 1353–1360, 2018.

[3] N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson. Modeling magnetic fields using Gaussian processes. In *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3522–3526. IEEE, 2013.

[4] A. Solin, S. Särkkä, J. Kannala, and E. Rahtu. Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning. In *2016 European Navigation Conference (ENC)*, pages 1–9. IEEE, 2016.

[5] B. Li, T. Gallagher, A. G. Dempster, and C. Rizos. How feasible is the use of magnetic field alone for indoor positioning? In *Proceedings of the 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9. IEEE, 2012.

[6] Frassl M. Angermann M. Doniec M. Julian B.J. Puyol M.G. Khider M. Lichtenstern M. Robertson, P. and L. Bruno. Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments. In *Proceedings of the International Conference on indoor positioning and indoor navigation*, pages 1–10. IEEE, 2013.

[7] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning. Simultaneous localization and mapping using ambient magnetic field. In *Proceedings of the 2010 IEEE Conference on Multisensor Fusion and Integration*, pages 14–19. IEEE, 2010.

[8] F. Viset, R. Helmons, and M. Kok. An extended kalman filter for magnetic field SLAM using Gaussian process regression. *Sensors*, 22(8):2833, 2022.

[9] W. Storms, J. Shockley, and J. Raquet. Magnetic field navigation in an indoor environment. In *Proceedings of the 2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*, pages 1–10. IEEE, 2010.

[10] C.K.I. Williams and C.E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[11] I. Skog. Inertial and magnetic-field sensor arrays - capabilities and challenges. In *Proceedings of the 2018 IEEE SENSORS*, pages 1–4, 2018.

[12] A. McHutchon. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, Citeseer, 2015.

[13] A. C. Damianou, M. K. Titsias, and N. D Lawrence. Variational inference for uncertainty on the inputs of Gaussian process models. 2014.

[14] A. Girard, C. Rasmussen, J.. Q. Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. *Advances in neural information processing systems*, 15, 2002.

[15] J. Q. Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. In *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 2, pages II–701. IEEE, 2003.

[16] P. Goldberg, C. Williams, and C. Bishop. Regression with input-dependent noise: A Gaussian process treatment. *Advances in neural information processing systems*, 10, 1997.

[17] H. J. Bijl. *LQG and Gaussian process techniques: For fixed-structure wind turbine control*. PhD thesis, Delft University of Technology, 2018.

[18] A. Girard and R. Murray-Smith. Learning a Gaussian process model with uncertain inputs. *Department of Computing Science, University of Glasgow, Tech. Rep. TR-2003-144*, 2003.

[19] A. McHutchon and C.E. Rasmussen. Gaussian process training with input noise. *Advances in Neural Information Processing Systems*, 24, 2011.

[20] I. Skog, G. Hendeby, and F. Gustafsson. Magnetic odometry-a model-based approach using a sensor array. In *Proceedings of the 2018 21st International Conference on Information Fusion (FUSION)*, pages 794–798. IEEE, 2018.

[21] M. Kok and T. B. Schön. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, 16(14):5679–5689, 2016.